IOWA STATE UNIVERSITY
Digital Repository

Retrospective Theses and Dissertations | Iowa State University Capstones, Theses and Dissertations

1998

Steganography using the Minimax Eigenvalue Decomposition

Chaka Allen
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Data Storage Systems Commons, and the Other Computer Engineering Commons

Recommended Citation
Allen, Chaka, "Steganography using the Minimax Eigenvalue Decomposition" (1998). *Retrospective Theses and Dissertations*. 17001.
https://lib.dr.iastate.edu/rtd/17001

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

المنارة للاستشارات

www.manaraa.com

Steganography using the Minimax Eigenvalue Decomposition

by

Chaka Allen

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Major Professor: Dr. Jennifer Davidson

Iowa State University

Ames, Iowa

1998

Copyright © Chaka Amir Allen 1998. All rights reserved.

www.manaraa.com

Graduate College
Iowa State University

This is to certify that the Master's thesis of

Chaka Amir Allen

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

www.manaraa.com

TABLE OF CONTENTS

المنارة للاستشارات

www.manaraa.com

ACKNOWLEDGEMENTS

I would first like to thank my Lord and Savior Jesus Christ for being with me throughout this entire process. Without Him, I would have never completed this thesis.

I would like to thank my major professor Dr. Jennifer Davidson. The freedom I was given to pursue this area was invaluable. Her standard of excellence and pursuit of the best in me, the best in my work, and the best in her own work has been a large source of motivation for me. I love to see excellence and she exemplifies that in her career.

Finally, I would like to thank the Department of Electrical and Computer Engineering at Iowa State University for funding my graduate study, as well as the Office of Naval Research (grant number N00014-93-1-0001). I hope to show that their dedication to my success was well worth it. Steganography is an area of research that very few professors had heard of before my research. I hope to bring some attention to the area through my thesis work.

www.manaraa.com

ABSTRACT

Steganography, or information hiding, is the study of methods to hide data within a medium. Handwritten communications, digital images, digital video and digital music are examples of various media in which data can be hidden. A primary application of steganography is the "watermarking" of digital information. Just as paper money, governmental documents, and even official school transcripts contain watermarks to prevent unauthorized reproduction, vendors seek to watermark their digital information to prevent piracy.

Methods of steganography can also be used to create *covert channels*. A covert channel is a public medium used to transmit private information. Data can be embedded using a steganographic algorithm in order to transmit private information rather than authenticate ownership. Media such as image data, sound data or even the access times of a file, can be used as covert channels.

Minimax algebra is known to have applications to machine scheduling, operations research, and image algebra. Raymond Cuninghame-Green provided the groundbreaking work in this area. Minimax algebra provides parallels to linear algebra by extending the real numbers and introducing new operations with amazing properties. Ritter and Sussner use minimax algebra as the mathematical foundation for a new image transform called the *minimax eigenvalue decomposition* (MED). This transform avoids computational difficulties encountered with other techniques used for image layering, compression and transmission.

المنارة للاستشارات

www.manaraa.com

This thesis presents research where properties of the MED transform are used to produce a flexible, computationally robust technique for hiding data within digital images. This new technique can be used to watermark an image or to convert an image into a covert channel. A measure is provided to determine how close an image containing message data is to the original image. Inherent in the technique are *keys,* or information separate from the message data stored in the image, that can establish authenticity of the image data, making this technique different from most steganography techniques that rely on embedded data integrity to establish authenticity. The technique is applied to ten different real images. An analysis of the results and ideas for future research are presented.

www.manaraa.com

INTRODUCTION

Throughout history, people have been concerned with the protection of information from unauthorized use or access. Information can be dangerous, or it can be helpful, depending on the intent of the user. The danger of unauthorized use or access, either real or perceived, has motivated people to find ways to protect or restrict either the distribution of information or the transaction of information from one place to another. However, methods of subverting restrictions and bypassing protections have emerged along with methods of protection and restriction.

Steganography, cryptography, and covert channels are three techniques used both to protect information and to subvert restrictions on the transaction of information. *Steganography*, or "covered writing", is the art and science of hiding information within a medium that is not hidden. Examples of such media are digital image data, analog or digital sound data, and hand-written or typed text. When used to pass information, these media can be referred to as *covert channels* [1]. *Digital watermarking* [3], for instance, is an application of steganography where digital information is placed within data to allow an owner to verify its authenticity. *Authentication* [16], or the establishment of ownership of information by a party, is an important application of digital watermarking. The authenticity of information is established by either making the embedded information or the algorithm for inserting or removing the information unique to the owner. Sometimes the uniqueness of an algorithm is established through the existence of *keys*, which are specific pieces of information necessary to recover the hidden information that can be tied both to the owner of information and to the information

المنارة للاستشارات www.manaraa.com

itself. *Cryptography* [16], on the other hand, is concerned with the creation and retrieval of scrambled information. Protection from unauthorized access is dependent upon the strength of the algorithm used to scramble the information. One use of cryptography is the creation and interpretation of *digital signatures* [16], or visible data created with the sole purpose of verifying the identity of a user and relating that user to a particular piece of information, like a contract for instance. The difference between a digital signature and a digital watermark is a digital watermark is designed to protect the author of information, whereas a signature is intended to protect the receiver of information. The recipient of a document gains no advantage from the removal of a digital signature [9].

Steganography, cryptography, and covert channels have been used throughout history. In World War II, American prisoners of war would conceal information from enemy censors by encoding a secret message in Morse code using the sequence of dots to the letter "i" and crosses to the letter "t" to communicate their message [15]. During the SALT II treaty talks between the United States and the U.S.S.R, an agreement was reached to place sensors on missile silos that would provide a count of the number of missiles each country had in their possession. However, the U.S. had a rotating missile system in their silos, used to shift the locations of their missiles, which they did not want revealed. Great concern was given to the analysis of those sensors for possible covert channels through which the U.S.S.R. could obtain information regarding the locations of missiles [15]. Another example is found in ancient Greece, where messages were sometimes written on the shaved scalp of a messenger and the hair allowed to grow back [2]. Today, cable television signals are scrambled to allow access to paid subscribers with the right unscrambling equipment, and encrypted versions of passwords on

www.manaraa.com

computer systems are stored instead of actual passwords to protect from hackers who may gain unauthorized access to the file.

In recent times, methods have been developed for commercial use to protect information. One method of steganography for images breaks an image into rectangles of pixels, embedding a mark or information in particular rectangles that follow the detail of the image [7]. A method of steganography for sound spreads info across frequencies of music data and uses that information to identify the owner of a musical selection [6]. A method called Patchwork [2] takes advantage of statistical properties of an image by embedding an imperceptible, specific statistic (one that has a Gaussian distribution) within a digital image. Another method takes the frequency transform of an image, imbeds a watermark, and performs the inverse frequency transform to produce an image containing hidden data [3].

Researchers are searching for steganography techniques that can provide *self-authentication*. Self-authentication is the ability of data to be authenticated without third party intervention. No method of steganography has been found to provide self-authentication. In other words, no method exists that cannot be rendered useless either by corrupting embedded data or by placing a second mark over an already embedded mark, making the first mark unreadable. These attacks are thwarted by the existence of third parties that can verify the previous existence of a mark. However, methods used by third party companies like Digimarc©, ARIS Technologies©, and MediaSec Technologies LLC© do not claim to be self-authentication methods, as no method has yet to be proven to be such.

المنارة للاستشارات
www.manaraa.com

Along with steganography and cryptography, covert channels and their analysis have found applications in computer systems. The Orange Book is a governmental publication that gives guidelines for evaluating the security of computer systems. In order for a system to be classified as an A1 system, which is the highest classification level for any system, there must be a formal analysis of the potential for covert communications in the system. To see how easily covert communication can be achieved, we consider an example given in [10]. Suppose two people, High and Low, of different security levels, each have access to an interactive chess game. Furthermore, suppose that if Low plays by himself, the response time is 1ms, an if Low plays with someone, the response time is 2ms. High can time when he plays with Low and when he doesn't, assuming that Low plays continually of course, to create a two letter alphabet through which he can communicate with Low. The first letter would be the 1ms response time, and the second letter would be the 2ms response time. However, according to the *-property of the Bell-Lapadula security model [14], a person or entity of a certain security clearance cannot communicate information to a person or an entity of a lower security clearance. Therefore, if this communication were allowed to take place, a breach in the security policy of the system would result.

The type of channel discussed above is called a *timing channel*. Another type of channel is a storage channel. A *storage channel* is used to send information to a receiver, where the receiver samples data at certain time intervals or spatial locations agreed upon previously. The person receiving the communication doesn't necessarily know the message he or she will receive, but he or she knows the message has to be complete by the time the communication is complete. A stream cover is a storage channel where a

www.manaraa.com

continuous data stream is used as a medium of communication [1]. Samples of the data are used to represent the message. Digital sound data, for example, can be used as a stream cover. A stream cover, however, cannot be used if the recipient wants to access the hidden data at will. A random access cover is a storage channel where the message can be accessed at will. An example of a random access cover is a frame on a Kodak Photo CD [18]. A frame's maximum resolution is 2048 by 3072 pixels, with each pixel containing 24 bits of information. By using only the least significant bit of each pixel to store a 0 or 1, a 2.3 megabyte message can be hidden in a single picture.

This thesis focuses on hiding information within digital images. The method uses a radical image transform called the Minimax Eigenvalue Decomposition (MED), which decomposes an image into layers without the computational and roundoff penalties encountered in the Singular Value Decomposition (SVD) [12]. The SVD is well-documented in matrix and image algebra literature, and is known to be computationally intensive due to the difficulty of computing eigenvalues and eigenvectors. The MED transform avoids these difficulties. This thesis utilizes the MED transform by combining message data and a subset of the layers produced by the transform to create an image that is "close" to the original image. The algorithm for the combining of the data and the selected layers yields keys useful for authenticating an image, even if the embedded data is tampered with.

This algorithm is believed to be weak if an image is to be used as a covert channel. However, this algorithm is believed to be strong if one wants to prove ownership of an image. Because the keys produced by the algorithm are not dependent on the data stored by the algorithm, authentication can be accomplished even if the data is

www.manaraa.com

tampered with, provided a trusted record of the keys exists. Although gray-value images are used in this thesis, the algorithm is based on a matrix transform, making it useful for any image converted into a matrix of pixel values, including color images. The amount of space available for storing a hidden message using this algorithm is dependent on the image used and the data stored.

Chapter 1 focuses on definitions and mathematical preliminaries necessary to describe the MED transform and the steganography concepts. Chapter 2 contains the discussion of the transform and its comparison to the SVD method. Chapter 3 contains the proof that motivated the data-hiding algorithm as well as the data-hiding algorithm itself. Chapter 4 contains examples of images processed by the algorithm. Chapter 5 contains an analysis and discussion of the results, and conclusions and ideas for future research with this algorithm conclude the thesis.

www.manaraa.com

CHAPTER 1: PRELIMINARIES AND DEFINITIONS

Minimax algebra serves as the mathematical foundation for this thesis, and [4] provides the groundbreaking work in the area. Initially applied towards job scheduling, this field of mathematics has been applied to the area of *image algebra* as well. Presented here are results from minimax algebra that are relevant to this thesis. Although the mathematics is not necessary to perform the algorithm, it is necessary for understanding how the algorithm works. This chapter contains a short presentation of definitions in steganography, and a brief introduction to minimax algebra.

1.1 Steganography terminology

Authentication:	Establishment of ownership of information. Information is *authenticated* if it can be proven to have originated from a particular source.
Attack:	Any manipulation of a medium that seeks to either destroy the information stored in that medium or the ability to authenticate the medium.
Message data or *Mark*:	Data stored within or passed via a digital medium. An image is an example of such a medium.
Stego image:	An image containing message data.
Key:	Information used to either retrieve data from a stego image or to authenticate an image.
Covert channel:	A public medium used to communicate private information

المنارة للاستشارات

www.manaraa.com

between parties. Sound data, image data, or even chess games, as discussed in the Introduction, are examples of media used as covert channels.

1.2 Minimax algebra: An introduction

Minimax algebra is a matrix-structured algebra that manipulates matrices and vectors with algebraic operations that are similar to but distinct from regular linear algebra. The general idea of minimax algebra is as follows. Take the usual linear algebra operation of matrix-matrix multiply:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj},$$

where $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are appropriately sized matrices. Now, replace the multiplication operation with addition, and replace the sum operation with a maximum operation, to get

$$\hat{c}_{ij} = \bigvee_{k=1}^{n} (a_{ik} + b_{kj}). \qquad (1)$$

This yields a matrix

$$\hat{C} = \{c_{ij}\}$$

which is in general much different from $C = \{c_{ij}\}$. The example given in Eq. (1) is a particular case that occurs in minimax algebra, and will be used in this thesis. This matrix algebra, when endowed with appropriate properties, produces a rich matrix structure which has continued concepts of eigenvalues and eigenvectors, solutions to

www.manaraa.com

systems of equations, matrix rank, spectral inequalities, and many other concepts parallel to those found in linear algebra.

We shall give only a brief introduction to minimax algebra as concerns the concepts in this thesis, and refer the reader to [4], [11], and [12] for further details.

A *semigroup* is defined as a set S with an operation $\oplus$ such that the following two properties hold:

G1: $x \oplus (y \oplus z) = (x \oplus y) \oplus z \; \forall \, x, y, z \in S$ (associativity)

G2: $x \oplus y = y \oplus x \; \forall \, x, y \in S$ (commutativity)

provided, of course, that S is closed under $\oplus$.

A *semiring* is defined as a semigroup $(S,\oplus)$ with an additional operation $\otimes$ satisfying the following properties:

R1: $x \otimes (y \otimes z) = (x \otimes y) \otimes z \; \forall \, x, y, z \in S$ (associativity of $\otimes$)

R2: $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z) \; \forall \, x, y, z \in S$ (left distributivity of $\otimes$)

R3: $(y \oplus z) \otimes x = (y \otimes x) \oplus (z \otimes x) \; \forall \, x, y, z \in S.$ (right distributivity of $\otimes$)

Let $M_{m\,x\,n}(S)$ be the set of m x n matrices whose entries are from S. The set $M_{m\,x\,n}(S)$ with the operation $\oplus$ is a semigroup as well, with $\oplus$ being defined in the following way:

$$A \oplus B = \{c_{ij}\}_{m\,x\,n}, \text{ where } c_{ij} = a_{ij} \oplus b_{ij}, \; \forall \; i \in \{1 \ldots m\}, j \in \{1 \ldots n\}. \quad (2)$$

$(S, \otimes)$ may be a semigroup as well; in which case, $(M_{m\,x\,n}(S), \otimes)$ is also a semigroup with $\otimes$ being defined by replacing $\oplus$ in Eq. (2) with $\otimes$.

A matrix product $\boxed{\otimes}$ can also be defined in the following way:

$$\boxed{\otimes} : M_{m\,x\,k}(S) \text{ x } M_{k\,x\,n}(S) \rightarrow M_{m\,x\,n}(S), \text{ and } A \boxed{\otimes} B = C, \text{ for } A \in M_{m\,x\,k}(S),$$

www.manaraa.com

$B \in M_{k\times m}(S)$, and $C \in M_{m\times n}(S)$, where

$$c_{ij} = \Gamma_{l=1}^{k}(a_{il} \oplus b_{lj}) = (a_{i1} \oplus b_{1j}) \otimes \ldots \otimes (a_{ik} \oplus b_{kj}).$$

If $S = \mathbf{R}$, $\oplus = \times$, $\otimes = +$, we have matrix multiplication as in linear algebra:

$$c_{ij} = \sum_{l=1}^{k} a_{il} \times b_{lj}.$$

Let us denote the set $\mathbf{R} \cup \{-\infty\}$ as $\mathbf{R}_{-\infty}$ and the set $\mathbf{R} \cup \{+\infty\}$ as $\mathbf{R}_{+\infty}$. Furthermore, let us define the operations $+$, $+'$, $\vee$ and $\wedge$ as follows:

$$a + b = \begin{cases} a + b, & a,b \in \mathbf{R} \\ -\infty, & a = -\infty \; or \; b = -\infty \end{cases}$$
$$a +' b = \begin{cases} a + b, & a,b \in \mathbf{R} \\ +\infty, & a = +\infty \; or \; b = +\infty \end{cases}$$
$$a \vee b = \begin{cases} a, & \text{if } a \geq b \\ b, & \text{otherwise.} \end{cases}$$
$$a \wedge b = \begin{cases} a, & \text{if } a \leq b \\ b, & \text{otherwise.} \end{cases}$$

Call $a \vee b$ the *max* operation, and call $a \wedge b$ the *min* operation.

If $S = \mathbf{R}_{-\infty}$, $\oplus = +$, $\otimes = \vee$, then $\boxed{\otimes} = \boxed{\vee}$, and the matrix product above takes the form:

$$A \boxed{\vee} B = C, where$$
$$c_{ij} = \bigvee_{l=1}^{k} a_{il} + b_{lj} = (a_{i1} + b_{1j}) \vee \ldots \vee (a_{ik} + b_{kj}).$$

We call this product a *max product.*

www.manaraa.com

Similarly, if $S = \mathbf{R}_{+\infty}$, $\oplus = +'$, $\otimes = \wedge$, then $\boxed{\otimes} = \boxed{\wedge}$, and the matrix product above takes the form:

$$A \boxed{\wedge} B = C, where$$
$$c_{ij} = \bigwedge_{l=1}^{k} a_{il} +' b_{lj} = (a_{i1} +' b_{1j}) \wedge \ldots \wedge (a_{ik} +' b_{kj}).$$

We call this product a *min product*.

Note: Boxes will be put around any special matrix operations, and will be left off of pointwise operations.

Furthermore, define A^* as the *negative transpose* of A. That is, if $A = \{a_{ij}\}$, then $A^* = \{a_{ij}^*\}$, where $a_{ij}^* = -a_{ji}$. The negative transpose of A will be essential to calculating the MED transform of A.

Let us define pointwise image addition, subtraction, multiplication, max, and min of two images $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$as follows, reiterating the note above.

$$A + B = C = \{c_{ij}\},\ c_{ij} = a_{ij} + b_{ij},$$

$$A - B = C = \{c_{ij}\},\ c_{ij} = a_{ij} - b_{ij},$$

$$A * B = C = \{c_{ij}\},\ c_{ij} = a_{ij} * b_{ij},$$

$$A \vee B = C = \{c_{ij}\},\ c_{ij} = a_{ij} \vee b_{ij},$$

$$A \wedge B = C = \{c_{ij}\},\ c_{ij} = a_{ij} \wedge b_{ij}.$$

Finally, let us define the *indicator threshold* T of A and the *indicator threshold image* $X_{\geq T}(A)$ as follows:

$$X \geq T(A) = C = \{cij\}, cij = \begin{cases} 1 & \text{if } aij \geq T \\ 0 & \text{otherwise.} \end{cases}$$

These notations are used to describe the storage algorithm.

www.manaraa.com

The semirings $(\mathbf{R}_{-\infty}, \vee, +)$ and $(\mathbf{R}_{+\infty}, \wedge, +')$ can be combined into what is known as the *bounded lattice ordered group (BLOG)* $(\mathbf{R}_{\pm\infty}, \vee, \wedge, +, +')$, where $\mathbf{R}_{\pm\infty} = \mathbf{R} \cup \{-\infty, +\infty\}$, and each operation is as defined above, resulting in a *minimax algebra*.

BLOGs have rich and interesting properties with a wide variety of applications. This thesis does not come close to exploring the numerous applications of minimax algebra to the areas of scheduling theory and image algebra ([4], [5]). This thesis provides another contribution to the area and more attention to the area's potential.

www.manaraa.com

CHAPTER 2: MINIMAX EIGENVALUE DECOMPOSITION (MED)

Dividing an image into layers for transmission or storage purposes is not new. The singular value decomposition (SVD) is well known, and has been used for this when applicable. The SVD, however, has computational difficulties that cannot always be overcome if accuracy or computational efficiency is a concern. In [12] there is an alternative to those computational difficulties through the use of a radical, new transform: the minimax eigenvalue decomposition (MED). Much of the mathematical foundations for the MED can be found in [4], and discussions to image processing applications can be found in [5], [11], and [17].

A discussion of the MED transform is presented and compared to the SVD of a matrix. The discussion of the SVD is used first as an example of familiar material before we move into the unfamiliar territory of the MED. The MED discussion is a presentation of work done in [12].

2.1 The SVD transform

The *singular value decomposition* (SVD) of A, a real-valued, m x n matrix (m≥n), provides a method for representing a matrix A as the product of three matrices as follows: $A=UDV^t$, U, V orthogonal m x m and n x n matrices, respectively, and

$$D=\begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & d_n \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix},$$

المنارة للاستشارات

www.manaraa.com

where d_1 through d_n are the square roots of the eigenvalues of the matrix A^tA, or the *singular values* of A. Furthermore, these singular values are ordered such that

$$d_1 \geq d_2 \geq \ldots \geq d_n.$$

The n columns of U, denoted $\mathbf{u}^1, \ldots, \mathbf{u}^n$, are the eigenvectors of AA^t, and the column vectors $\mathbf{v}^1$ through $\mathbf{v}^n$ of V are the eigenvectors of A^tA.

The SVD has a well-known application to image compression and transmission. The matrix A can be written as

$$A = \sum_{i=1}^{r} d_i \mathbf{u}^i (\mathbf{v}^i)^t,$$

where r represents the rank of D and $(\mathbf{v}^i)^t$ denotes the transpose of the vector $\mathbf{v}^i$. Image compression can be achieved due to the fact that

$$A \approx \sum_{i=1}^{k} d_i \mathbf{u}^i (\mathbf{v}^i)^t$$

where k << r. This allows us to store or transmit only the set

$$\{d_i, \mathbf{u}^i, \mathbf{v}^i\}_{i=1}^{k}$$

if we want to represent the image with less data.

<u>2.2 The MED transform</u>

An element $\lambda \in \mathbf{R}_{\pm\infty}$ is a *minimax eigenvalue* of A with corresponding *eigenvector* $\mathbf{x} \in (\mathbf{R}_{\pm\infty})^m$, if

$$A \boxed{\vee} \mathbf{x} = \lambda + \mathbf{x}.$$

www.manaraa.com

Here, $\lambda + \mathbf{x} \equiv (\lambda + x_1, \lambda + x_2, \ldots, \lambda + x_m)$, where $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, and $A \in$ $M_{m \times m}(\mathbf{R}_{\pm\infty})$.

Now, the matrix $U = A \boxed{\wedge} A^*$ has been proven [4] to have the property that each column of A is an eigenvector of U corresponding to eigenvalue 0. In other words, $U \boxed{\vee} \mathbf{a}^{\mathrm{i}} = \mathbf{a}^{\mathrm{i}}$, where $\mathbf{a}^{\mathrm{i}}$ represents the i-th column of A, and hence $U \boxed{\vee} A = A$.

The MED of an m x m matrix A can be written as

$$A = U \boxed{\vee} D \boxed{\vee} V^t,$$

where $V = A^t$,

$$D = \begin{bmatrix} d_1 & -\infty & \cdots & -\infty \\ -\infty & d_2 & \cdots & -\infty \\ \vdots & \vdots & \cdots & \vdots \\ -\infty & -\infty & \cdots & d_m \end{bmatrix},$$

and $d_1 = d_2 = \ldots = d_m = 0$.

This transform avoids inaccuracies in eigenvalue computation since every eigenvalue is zero.

Furthermore, we can represent A as a composition of the matrices $\mathbf{u}^{\mathrm{i}} \boxed{\vee} (\mathbf{v}^{\mathrm{i}})^{\mathrm{t}}$:

$$A = \bigvee_{i=1}^{m} d_i + \mathbf{u}^{\mathrm{i}} \boxed{\vee} (\mathbf{v}^{\mathrm{i}})^t = \bigvee_{i=1}^{m} \mathbf{u}^{\mathrm{i}} \boxed{\vee} (\mathbf{v}^{\mathrm{i}})^t,$$

where $(\mathbf{v}^{\mathrm{i}})^t = \mathbf{a}_{\mathrm{i}}$, if $\mathbf{a}_{\mathrm{i}}$ represents the i-th row of A. Hence, the above equation changes to:

$$A = \bigvee_{i=1}^{m} \mathbf{u}^{\mathrm{i}} \boxed{\vee} \mathbf{a}_{\mathrm{i}}.$$

www.manaraa.com

We call each $\mathbf{u}^{\mathbf{i}} \boxed{\vee} \mathbf{a}_{\mathbf{i}}$ matrix a *layer* of A. Each $\mathbf{u}^{\mathbf{i}} \boxed{\vee} \mathbf{a}_{\mathbf{i}}$ pair is an m x m matrix, and the pointwise max operation of all m layers is equivalent to the original m x m matrix A. These are analagous to the $d_i\mathbf{u}^{\mathbf{i}}(\mathbf{v}^{\mathbf{i}})'$ elements in the SVD transform discussion. We shall call these elements layers as well.

In the SVD transform, it is trivial to obtain the best approximating subset of k layers for a given $k \leq n$ after the difficult task of obtaining the eigenvalues and eigenvectors has been accomplished. In the MED transform, there is no eigenvalue computation since they are all zero, and the U matrix computation is trivial. The determination of a set of the k best approximating layers for a given k, however, is not trivial. The measure that follows provides the basis for finding k best layers. Hence, the accuracy of reproduction is not a concern. The following algorithm is discussed in [12].

Define a measure s(A,B) as follows:

$$s(A,B) = \sum_i \sum_j c_{ij}, \textit{ where}$$

$$c_{ij} = \begin{cases} a_{ij} - b_{ij}, & a_{ij} > b_{ij}, \\ 0 & \textit{otherwise.} \end{cases}$$

Now, for the ordering process, we have this algorithm:

<u>Ordering algorithm</u>

1) Find $\mathbf{u}^{\mathbf{i}} \boxed{\vee} \mathbf{a}_{\mathbf{i}} = \min(s(A, \mathbf{u}^{\mathbf{i}} \boxed{\vee} \mathbf{a}_{\mathbf{i}}), 1 \leq i \leq m)$. Name this pair $\mathbf{u}^{\mathbf{1}}$ and $\mathbf{a}_{\mathbf{1}}$.

2) Suppose you have found the first j ($j \geq 1$) eigenvector pairs. Now, find $\mathbf{u}^{\sigma}$ and $\mathbf{a}_{\sigma}$ such that:

$$\mathbf{u}^{\sigma} \boxed{\vee} \mathbf{a}_{\sigma} = max[s(\mathbf{u}^{i} \boxed{\vee} \mathbf{a}_{i}, \vee_{\mathrm{k=1}}^{\mathrm{j}} (\mathbf{u}^{\sigma k} \boxed{\vee} \mathbf{a}_{\sigma k})), 1 \leq i \leq m].$$

www.manaraa.com

Rename this pair as $\mathbf{u}^{j+1}$ and $\mathbf{a}_{j+1}$. Furthermore, let us define A_j as follows:

$$Aj = \bigvee_{k=1}^{j} (\mathbf{u}^{\sigma k} \boxed{\vee} \mathbf{a}_{\sigma k}).$$

We call A_j the *jth reconstruction* of A.

3) Set j = j + 1 and repeat steps 2 and 3 until j = m.

In addition, it is straightforward to show that

$$A_j \leq A$$

for each $j = 1, \ldots, m$. This information is crucial in producing our stegoimage, as we will see in Chapter 3. The next example illustrates the process discussed above.

2.3 Example 1: Layer computation

Define a matrix A as follows:

$$A = \begin{pmatrix} 8 & 4 & 20 & 47 & 3 \\ 6 & 56 & 48 & 0 & 8 \\ 27 & 88 & 13 & 97 & 65 \\ 77 & 17 & 0 & 0 & 39 \\ 53 & 17 & 47 & 67 & 0 \end{pmatrix}.$$

Then, we compute the $\mathbf{u}^1 \boxed{\vee} \mathbf{a}_1$ layer as follows:

$$A^* = \begin{pmatrix} -8 & -6 & -27 & -77 & -53 \\ -4 & -56 & -88 & -17 & -17 \\ -20 & -48 & -13 & 0 & -47 \\ -47 & 0 & -97 & 0 & -67 \\ -3 & -8 & -65 & -39 & 0 \end{pmatrix},$$

المنارة للاستشارات

www.manaraa.com

$$U = \begin{pmatrix} 0 & -52 & -84 & -69 & -45 \\ -47 & 0 & -97 & -71 & -67 \\ -7 & -35 & 0 & -50 & -34 \\ -47 & -48 & -97 & 0 & -67 \\ -3 & -39 & -71 & -39 & 0 \end{pmatrix}, \text{and}$$

$$\mathbf{u}^1 \boxed{\vee} \mathbf{a}_1 = \begin{pmatrix} 0 \\ -47 \\ -7 \\ -47 \\ -3 \end{pmatrix} \boxed{\vee} \begin{pmatrix} 8 & 4 & 20 & 47 & 3 \end{pmatrix} = \begin{pmatrix} 8 & 4 & 20 & 47 & 3 \\ -39 & -43 & -27 & 0 & -44 \\ 1 & -3 & 13 & 40 & -4 \\ -39 & -43 & -27 & 0 & -44 \\ 5 & 1 & 17 & 44 & 0 \end{pmatrix}.$$

Now, if we compute all 5 layers, and then compute the measure in step 1 of our algorithm above for each layer, we get these results:

$$s(A, \mathbf{u}^1 \boxed{\vee} \mathbf{a}_1) = 917$$

$$s(A, \mathbf{u}^2 \boxed{\vee} \mathbf{a}_2) = 1087$$

$$s(A, \mathbf{u}^3 \boxed{\vee} \mathbf{a}_3) = 1102$$

$$s(A, \mathbf{u}^4 \boxed{\vee} \mathbf{a}_4) = 1287$$

$$s(A, \mathbf{u}^5 \boxed{\vee} \mathbf{a}_5) = 952.$$

Therefore, $\mathbf{u}^1 \boxed{\vee} \mathbf{a}_1$ would be our $\mathbf{u}_{\sigma_1} \boxed{\vee} \mathbf{a}_{\sigma_1}$, or our A_1. Proceeding on with step 2 of our algorithm, we find:

$$s(\mathbf{u}^1 \boxed{\vee} \mathbf{a}_1, A_1) = 0$$

$$s(\mathbf{u}^2 \boxed{\vee} \mathbf{a}_2, A_1) = 393$$

$$s(\mathbf{u}^3 \boxed{\vee} \mathbf{a}_3, A_1) = 351$$

www.manaraa.com

$$s(\mathbf{u}^4 \boxed{\vee} \mathbf{a}_4, A_1) = 402$$

$$s(\mathbf{u}^5 \boxed{\vee} \mathbf{a}_5, A_1) = 199$$

Therefore, $\mathbf{u}^4 \boxed{\vee} \mathbf{a}_4$ is $\mathbf{u}^{\sigma 2} \boxed{\vee} \mathbf{a}_{\sigma_2}$, and:

$$A2 = (\mathbf{u}^{\sigma 1} \boxed{\vee} \mathbf{a}_{\sigma 1}) \vee (\mathbf{u}^{\sigma 2} \boxed{\vee} \mathbf{a}_{\sigma 2}) = \begin{pmatrix} 8 & 4 & 20 & 47 & 3 \\ 6 & -43 & -27 & 0 & -32 \\ 27 & -3 & 13 & 40 & -4 \\ 77 & 17 & 0 & 0 & 39 \\ 38 & 1 & 17 & 44 & 0 \end{pmatrix}.$$

If we repeat this process 3 more times, we will find that A_5 is indeed equivalent to our original matrix.

Note that, for any $\mathbf{u}^{\mathbf{i}} \boxed{\vee} \mathbf{a}_{\mathbf{i}}$ pair included in the A_j calculation, the s measure calculation between it and A_j must be zero. This is due to the fact that every element in A_j is greater than or equal to the corresponding element in the $\mathbf{u}^{\mathbf{i}} \boxed{\vee} \mathbf{a}_{\mathbf{i}}$ matrix. Hence, since step 2 looks for the maximum measure between the $\mathbf{u}^{\mathbf{i}} \boxed{\vee} \mathbf{a}_{\mathbf{i}}$ matrices and A_j for a particular n, duplication of layers found in this calculation is not a concern unless all the measure results equate to 0. If all the measure calculations in step 2 equate to 0, we are done, understanding that this reconstruction is equivalent to the original matrix.

Our focus will not be on simply reconstructing images. We want to store data within the reconstructions. The theorem discovered and proven in the next section provides a method to measure just how much space each reconstruction provides for data storage.

المنارة للاستشارات

www.manaraa.com

CHAPTER 3: PRODUCING A STEGOIMAGE

We now present the algorithm for producing a stego image. First, a theorem is presented that provides the foundation for the algorithm. Then, a general storage algorithm is presented that stores the message data within the image, producing a stego image.

3.1 Theorem

This theorem is instrumental in providing the user with a way to determine which pixel locations in the image can be used to store message data. We will use this theorem to aid in defense against attacks as well.

Theorem: Let A be an m x m matrix. Let A_j be the jth reconstruction of A. Then, A_j will contain at least j rows of A. More specifically, if the jth reconstruction of A is

$$A_j = \bigvee_{k=1}^{j} (\mathbf{u}^{\sigma_k} \boxed{\vee} \mathbf{a}_{\sigma_k}),$$

where $\mathbf{u}^{\sigma_k} \boxed{\vee} \mathbf{a}_{\sigma_k} = \mathbf{u}^{t_i} \boxed{\vee} \mathbf{a}_{t_i}$, $t_i \in \{1, 2, \ldots m\}$, $i \in \{1, 2, \ldots j\}$, then A_j will contain at least the rows $t_1, t_2, \ldots t_j$ of A.

In other words, we can track *precisely* where we have locations available for storage. Since $A_j \leq A$ (from Section 2.2), and we know that at least the j rows $t_1, \ldots, t_j$ of A_j are exactly the same as the j rows $t_1, \ldots, t_j$ of A, the remaining m-j rows of A_j potentially have individual pixel values that are less than or equal to A (since $A_j \leq A$). It is in these m-j rows that the storage algorithm, presented in Section 3.3, will try to hide the message data.

www.manaraa.com

Proof: Let $U=\{ u_{ij} \}$, $A=\{ a_{ij} \}$, and $A^{*}=\{ (a^{*})_{ij} \}$. Then, because $U=A \boxed{\wedge} A^{*}$, $u_{ij} = \min\{ a_{ik} + (a^{*})_{kj},\ 1 \le k \le m\}$, where $(a^{*})_{kj}$ is the element in the kth row and jth column of A^{*}. Now, $u_{ii} = \min\{ a_{ik} + (a^{*})_{ki},\ 1 \le k \le m\}$. But $(a^{*})_{ki} = -a_{ik}$. Therefore, $u_{ii} = 0$ for all i between 1 and m. Now, let $\mathbf{u}^{k} \boxed{\vee}\, \mathbf{a}_{k} = \{(b^{k})_{ij}\}$. This means that $(b^{k})_{ij} = u_{ik} + a_{kj}$ for all i, j, k between 1 and m. Now, the kth row of $\mathbf{u}^{k} \boxed{\vee}\, \mathbf{a}_{k}$ contains the elements $(b^{k})_{kj} = u_{kk} + a_{kj}$, $1 \le j \le m$. But, since $u_{kk} = 0$ for all k, $(b^{k})_{kj} = a_{kj}$ for all j. This means that the kth row of $\mathbf{u}^{k} \boxed{\vee}\, \mathbf{a}_{k}$ is equivalent to the kth row of A. Now, the element contained in the kth row and jth column in $\mathbf{u}^{n} \boxed{\vee}\, \mathbf{a}_{n}$, $n \neq k$, is $(b^{n})_{kj} = u_{kn} + a_{nj}$. This means that $(b^{n})_{kj} = a_{kv} + (a^{*})_{vn} + a_{nj}$ for some v corresponding to that element. If we can show that $(b^{n})_{kj} \le a_{kj}$ for any j, we can show that the kth row of $\mathbf{u}^{n} \boxed{\vee}\, \mathbf{a}_{n}$ is, per element, always less than or equal to the kth row of $\mathbf{u}^{k} \boxed{\vee}\, \mathbf{a}_{k}$. This is trivial, seeing as $a_{kv} + (a^{*})_{vn} + a_{nj} \le a_{kj} + (a^{*})_{jn} + a_{nj} \le a_{kj}$ for any j due to the $\boxed{\wedge}$ operation. This means that, if you perform the operation $(\mathbf{u}^{k} \boxed{\vee}\, \mathbf{a}_{k}) \vee (\mathbf{u}^{n} \boxed{\vee}\, \mathbf{a}_{n})$, both the kth and nth rows of this matrix will be equivalent to the kth and nth rows of A, since $(b^{k})_{kj} = a_{kj}\ \forall$ j from above. Furthermore, no other elements in this matrix will be greater than the corresponding element in A because no other elements in either $\mathbf{u}^{k} \boxed{\vee}\, \mathbf{a}_{k}$ or $\mathbf{u}^{n} \boxed{\vee}\, \mathbf{a}_{n}$ are greater than the corresponding elements in A. Since this argument can be extended to any max sum of the form

$$(\mathbf{u}^{t_1} \boxed{\vee}\, \mathbf{a}_{t_1}) \vee (\mathbf{u}^{t_2} \boxed{\vee}\, \mathbf{a}_{t_2}) \vee \dots \vee (\mathbf{u}^{t_n} \boxed{\vee}\, \mathbf{a}_{t_n}),$$

where it can be shown that this sum will produce a matrix that contains at least rows t_1, t_2, ... t_n of A, we have successfully proven our theorem.

المنارة للاستشارات

www.manaraa.com

We saw in Example 1 that A_2 contained rows 1 and 4 of A. In that example, A_2 contained *only* those rows of A because the layers used to create A_2 only contained those rows of A. However, if we look at the next example, we find a different result. This means that it is possible to have more rows of A included in the reconstruction than the t_1, ... t_n rows from $\mathbf{u}^{t_1} \boxed{\vee} \mathbf{a}_{t_1}$, ... $\mathbf{u}^{t_n} \boxed{\vee} \mathbf{a}_{t_n}$. This information is important to us. The more rows included from A, the less space there is to store data according to the algorithm that follows.

3.2 Example 2: All rows included in all layers

Let $A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix}$. Then,

$$A^* = \begin{pmatrix} -1 & -6 & -11 & -16 & -21 \\ -2 & -7 & -12 & -17 & -22 \\ -3 & -8 & -13 & -18 & -23 \\ -4 & -9 & -14 & -19 & -24 \\ -5 & -10 & -15 & -20 & -25 \end{pmatrix}, \text{and}$$

$$U = \begin{pmatrix} 0 & -5 & -10 & -15 & -20 \\ 5 & 0 & -5 & -10 & -15 \\ 10 & 5 & 0 & -5 & -10 \\ 15 & 10 & 5 & 0 & -5 \\ 20 & 15 & 10 & 5 & 0 \end{pmatrix}.$$

المنارة للاستشارات

www.manaraa.com

But this means that:

$$\mathbf{u}^1 \boxed{\vee} \mathbf{a}_1 = \begin{pmatrix} 0 \\ 5 \\ 10 \\ 15 \\ 20 \end{pmatrix} \boxed{\vee} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix} = A!$$

The same is true for all the layers $\mathbf{u}^k \boxed{\vee} \mathbf{a}_k, k \in \{1 \ldots 5\}$.

An explanation of this peculiar example can be found in [17], where *rank* of a matrix is defined. The rank of A above is 1, and thus any pair $\mathbf{u}^k \boxed{\vee} \mathbf{a}_k$, $k \in \{1,\ldots,5\}$, will satisfy $\mathbf{u}^k \boxed{\vee} \mathbf{a}_k = A$.

3.3 Storing message data within an image file

This section describes the storage algorithm for hiding data within an image reconstruction A_j. The particular storage algorithm we use traverses the set of appropriate rows in a raster-scan order to hide the data, that is, in a left-to-right order in the row, rows top to bottom in the image. The presentation of the general algorithm assumes a raster-scan order of traversal. However, any order of traversal is possible as long as the order can be recreated, so that the message data can be extracted properly.

We simulated message data by producing a string of 0s and 1s that was generated uniformly and randomly using the *rand*() function in C. This string simulates a well-

المنارة للاستشارات
www.manaraa.com

encrypted message that we may want to hide in an image. From cryptology, we know that encrypting an original message into one that has a uniform distribution of symbols makes it more difficult to decrypt the scrambled message. We assume that the message to be stored in the image is encrypted in such a manner because it provides an additional layer of security against possible attacks on the stego image.

This storage algorithm reads bits from a data file and adds them using integer addition to the pixel values at certain locations in the reconstruction image A_j, thereby creating a new image that hopefully is close to the original. This new image is the stego image. The pixel locations are chosen from certain rows of the images selected in the storage algorithm as described below. We also describe a measure that we use to evaluate how close the stego image is to the original image.

The main idea of the storage algorithm is to place pieces of the message data at these pixel locations in A_j where A_j is strictly less than the original image A. Recall that $A_j \leq A$ pointwise, although it is possible that $A_j = A$; see Example 2 in Section 3.2. By the theorem in Section 3.1, we know that the reconstructed image A_j will have at least j rows in it that are exactly equal to the corresponding j rows in the original image A. It is in the remaining m-j rows that A_j may be strictly less than A, and the storage algorithm searches those m-j rows.

In the description of the storage algorithm, we let $r_1, r_2, \ldots r_{m-j}$ label the m-j row numbers of A that are not used in the reconstructed image A_j. Denote an element in row r_n and column k of A_j by $(a_j)_{r_n k}$, and denote an element in in row r_n and column k of A by $a_{r_n k}$. Then by the Theorem, it is possible that $(a_j)_{r_n k}$ is strictly less than $a_{r_n k}$.

www.manaraa.com

Storage algorithm

1) Take an image A and compute A_j for some j.

2) Compute the image

$$A_j^+ = A_j * \mathrm{X}_{\geq 0}(A_j).$$

The indicator threshold is $T = 0$, and thus the image $A_j^+ = \{(a_j^+)_{hk}\}$ "zeroes out" the negative values in A_j.

3) Next compute the image

$$B = A - A_j^+.$$

Thus, $B = \{b_{hk}\}$, where

$$bhk = \begin{cases} ahk - (aj)hk & \text{if } (aj)hk \geq 0 \\ ahk & \text{if } (aj)hk < 0. \end{cases}$$

4) Compute the image $C = \{c_{hk}\}$, where

$$c_{hk} = \text{number of bits needed to represent } b_{hk}.$$

Note that if A has 255 gray values, then $0 \leq c_{hk} \leq 8$ for all h and k.

5) Let $D = C * \mathrm{X}_{\geq 0}(C - 1)$. Thus $D = \{d_{hk}\}$ satisfies

$$dhk = \begin{cases} chk - 1 & \text{if } chk \geq 1 \\ 0 & \text{if } chk = 0. \end{cases}$$

The image D represents the number of bits from the message datafile that will be stored in the corresponding pixel location in the image A_j^+. We have chosen to store one fewer bit per location than the maximum possible, for reasons discussed below. Thus, for example, if $d_{00} = 5$, then 5 bits from the message datafile will be

المنارة للاستشارات
www.manaraa.com

stored (by integer addition) at location (0,0) in A_j^+. From the Theorem, we know that nonzero values in D can only exist in rows $r_1, r_2, \ldots r_{m-j}$ of D.

6) Using a raster-scan order, we start at row r_1, and we find the first entry d_{r_1k} where $d_{r_1k} \geq 1$. Then, the first d_{r_1k} bits are pulled off of the message datafile, and we compute

$$(a_s)_{r_1k} = (a_j^+)_{r_1k} + p_{r_1k},$$

where p_{r_1k} is the base-10 representation of the first d_{r_1k} bits in the image file. The image $A_s = \{(a_s)_{hk}\}$ is the stego image. Then, the next entry d_{r_1m} where $d_{r_1m} \geq 1$ is found, and the next d_{r_1m} bits are pulled off of the message datafile, translated into a base-10 representation, and added to $(a_j^+)_{r_1m}$. This process is repeated for all the entries in all of the rows $r_1, r_2, \ldots r_{m-j}$ of D and A_j^+, until either the entire message datafile is stored, or we have traversed all available locations in the m-j rows for storage. If we run out of data, or if $d_{r_nk} = 0$ for some $n \in \{1\ldots m-j\}$, $k \in \{1\ldots m\}$, then

$$(a_s)_{r_nk} = a_{r_nk}.$$

That is, the steo image takes on the values of the original image at those particular locations.

7) Define $A_s = \{(a_s)_{hk}\}$ as follows:

$$(as)hk = \begin{cases} (as)rgk, & \text{if } h = rg, g \in \{1,2,m-j\} \\ ahk & \text{otherwise.} \end{cases}$$

We call A_s the *stego image*.

www.manaraa.com

Figure 3.1 gives a pictorial overview of the storage algorithm. We chose to store one less bit of information per appropriate pixel location than the maximum possible to avoid having to re-scale the stego image A_s after the procedure was completed. For example, if $a_{hk} = 250$, and $(a_j^+)_{hk} = 230$, then $b_{hk} = 250 - 230 = 20$, and $c_{hk} = 5$, since it takes 5 binary bits to store the decimal number 20. We could then chose to store the next 5 bits from our message file at location (h,k) of A_j^+. However, if the next 5 bits that were available in the message file were 11100_2 (the number 28 in base 10), then $(a_s)_{hk} = 230 + 28 = 258 > 255$. Since we are representing A_s by 8 bits only, we would have to re-scale A_s to ensure that pixel values fall within this range. Re-scaling could also change the visual appearance of A_s significantly. Reducing the number of bits stored by one avoids this situation. Since we want to make A_s visually close to A and preserve data integrity, we want to avoid re-scaling.

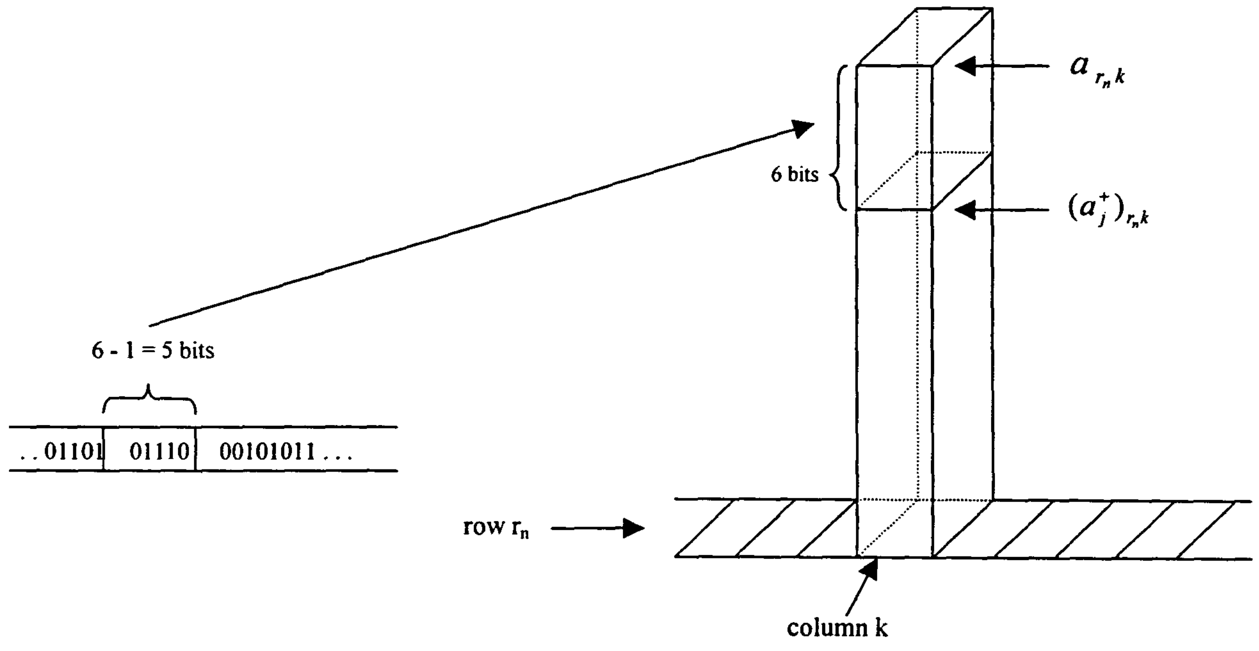


Figure 3.1. Picture of storage algorithm.

www.manaraa.com

We work with A_j^+ instead of A_j to avoid re-scaling as well. If, for example, $a_{hk} = 31$, and $(a_j)_{hk} = -31$, then $a_{hk} - (a_j)_{hk} = 62 = 111110_2$. This would mean that $d_{hk} = 6 - 1 = 5$. The next 5 bits from the datafile could range from the decimal values 0 to 31. However, this would mean that $(a_s)_{hk}$ could only range from $-31 + 0 = -31$ to $-31 + 31 = 0$, in which case it is most probable that $(a_s)_{hk}$ would contain a negative value to re-scale.

We also remark that a different order of visiting the sites marked for storage besides a raster-scan (corresponding to nonzero locations in D) will result in a different stego image A_s.

We define a measure between two images A and F as follows:

$$m(A,F) = \frac{\sum_{n=1}^{m-j}\sum_{k=1}^{m} a_{rnk} - (f)_{rnk}}{(m-j)\times m}.$$

Here, F is an image of type A_j (reconstruction) or A_s (stego). This measures the average difference per location used for storage between the original image and the image F. This is one measure we will use to determine how "close" our stego image is to our original image. The measure takes the differences between the image F and the original image at each matrix location, sums them, and divides the sum by the total number of elements in all valid rows. The closer the measure is to 0, the closer the image F is to our original image. Next we look at A and A_2 from Example 1.

www.manaraa.com

3.4 Example 3: Storage algorithm applied to an image

$$A=\begin{pmatrix}8&4&20&47&3\\6&56&48&0&8\\27&88&13&97&65\\77&17&0&0&39\\53&17&47&67&0\end{pmatrix}\qquad A_2=\begin{pmatrix}8&4&20&47&3\\6&-43&-27&0&-32\\27&-3&13&40&-4\\77&17&0&0&39\\38&1&17&44&0\end{pmatrix}$$

$$A_2^+=\begin{pmatrix}8&4&20&47&3\\6&0&0&0&0\\27&0&13&40&0\\77&17&0&0&39\\38&1&17&44&0\end{pmatrix}\qquad B=\begin{pmatrix}0&0&0&0&0\\0&56&48&0&8\\0&88&0&57&65\\0&0&0&0&0\\15&16&30&23&0\end{pmatrix}$$

$$C=\begin{pmatrix}0&0&0&0&0\\0&6&6&0&4\\0&7&0&6&7\\0&0&0&0&0\\4&5&5&5&0\end{pmatrix}\qquad D=\begin{pmatrix}0&0&0&0&0\\0&5&5&0&3\\0&6&0&5&6\\0&0&0&0&0\\3&4&4&4&0\end{pmatrix}$$

$$A_s=\begin{pmatrix}8&4&20&47&3\\6&0+5\text{ data bits}&0+5\text{ data bits}&0&0+3\text{ data bits}\\27&0+6\text{ data bits}&13&40+5\text{ data bits}&0+6\text{ data bits}\\77&17&0&0&39\\38+3\text{ data bits}&1+4\text{ data bits}&17+4\text{ data bits}&44+4\text{ data bits}&0\end{pmatrix}.$$

The three rows available for storing message data are $r_1 = 2$, $r_2 = 3$, and $r_3 = 5$. Using a raster-scan visiting order, the first valid location to check for possible storage of data is the (1,1) location of A_2, as $d_{11} = 5$. We would then read off 5 bits from our datafile, and

المنارة للاستشارات
www.manaraa.com

add the base 10 representation of those bits to $(a_2^+)_{11}$ to create $(a_s)_{11}$. We would follow similar procedures to find $(a_s)_{12}$, $(a_s)_{14}$, $(a_s)_{21}$, $(a_s)_{23}$, $(a_s)_{24}$, $(a_s)_{50}$, $(a_s)_{51}$, $(a_s)_{52}$, and $(a_s)_{53}$, until we either run out of bits to read or each of these locations has been visited. If we run out of bits, we fill the remaining pixel locations with the original image pixel values at those locations.

3.5 Extracting data from the stego image

Here we give a short description on how to recover data from a stego image.

In brief, the process above is inverted. The receiver of A_s will need to know A and the reconstruction value j, as well as the traversal order followed when the message data was originally inserted into A_j^+. The receiver computes A_j^+, knowing A and j, and computes the image D, following the procedure outlined in Section 3.3. Then the receiver pulls off the number of bits d_{hk} from each location (h,k) at the corresponding locations in A_s. Thus, for each (h,k) where $d_{hk} \neq 0$, the value $(a_s)_{hk} - (a_j^+)_{hk}$ is computed, and $(a_s)_{hk} - (a_j^+)_{hk}$ will be represented by d_{hk} bits. The bits are pulled off in the traversal order used to insert them, and the message data file consists of the bit string created by stringing those bits together in that order. Then decrypting of the message is performed if needed.

www.manaraa.com

CHAPTER 4: RESULTS

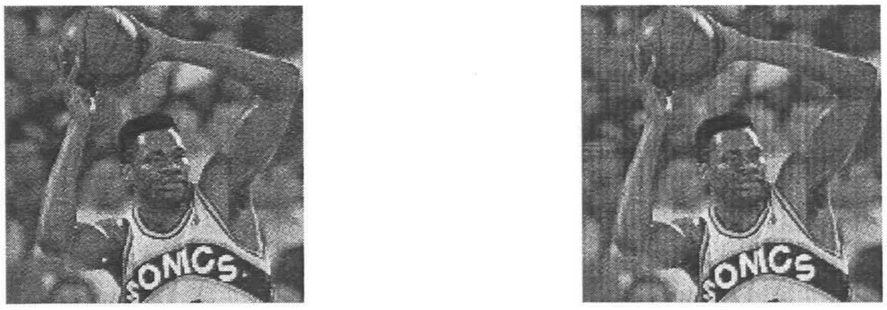
In this chapter we present the results of applying the storage algorithm given in Chapter 3 to ten different images. The images were obtained from web sites, and a wide variety of images were selected. We chose to set j = 125 for each of the seven images, although the (square) image sizes varied from 149 x 149 to 326 x 326. We chose j = 125 arbitrarily. The space available to hide data increases as j decreases. However, the smaller the value for j, the higher the possibility for error as less of the original image will be included within the reconstruction image A_j. The size of each image, the level of reconstruction, the number of locations available for storage, the difference measure base 10, the difference measure in bits, and the average L_1 error over the storage locations are included for each image in Table 4.1. The average L_1 error is computed over the number of storage locations only (not over (m-j) * m locations). The difference measure in the 5th column provides a value representing the average difference between the stego image and the original image per pixel location used to store message data. Figures 4.1 to 4.10 contain images to which the algorithm was applied. Each figure can be found in [19].

المنارة للاستشارات

www.manaraa.com

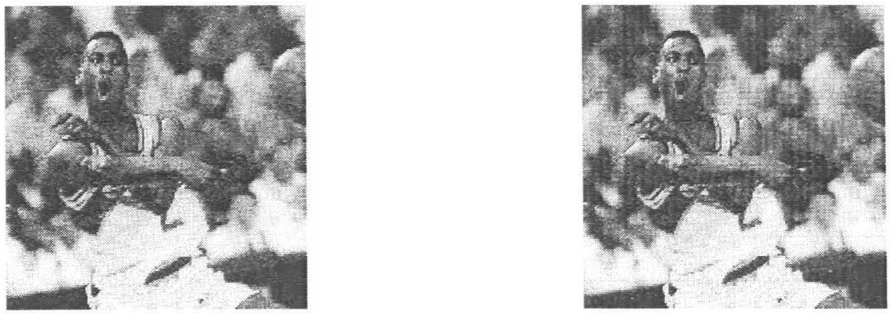
Table 4.1: Image information

Images	Size	j	# stor. lctns.	# bits stored	#bits/ lctn.	Diff. meas.	Diff. Meas. (bits)	L_1 error over storage lctn.
Figure 4.1	149 x 149	125	2395	9481	3.96	5.78	2.53	8.63
Figure 4.2	150 x 150	125	2750	11911	4.33	8.43	3.07	11.49
Figure 4.3	165 x 165	125	4711	14982	3.18	3.96	1.98	5.54
Figure 4.4	169 x 169	125	5014	15314	3.05	3.49	1.80	5.17
Figure 4.5	171 x 171	125	5805	19063	3.28	4.04	2.01	5.47
Figure 4.6	178 x 178	125	1749	2297	1.31	0.228	-2.13	1.22
Figure 4.7	195 x 195	125	9902	39828	4.02	10.8	3.43	14.89
Figure 4.8	215 x 215	125	11556	37461	3.24	3.59	1.84	6.01
Figure 4.9	263 x 263	125	28762	107686	3.74	7.30	2.86	9.21
Figure 4.10	326 x 326	125	58073	230084	3.96	9.02	3.17	10.18



a. Original image. b. Stego image.

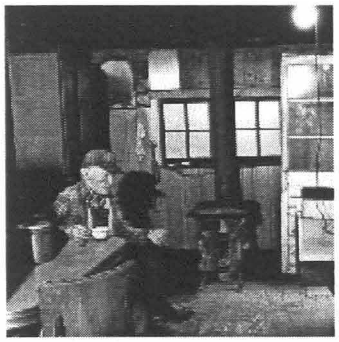
Figure 4.1. Basketball player 1.



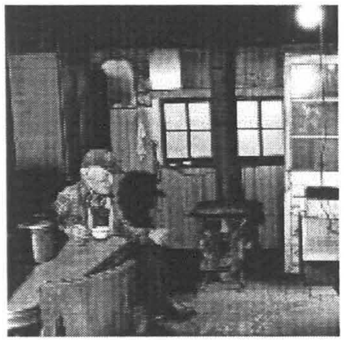
a. Original image. b. Stego image.

Figure 4.2. Basketball player 2.

المنارة للاستشارات
www.manaraa.com

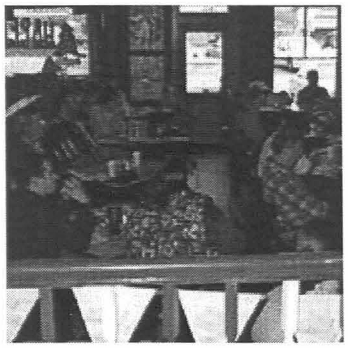


a. Original image.

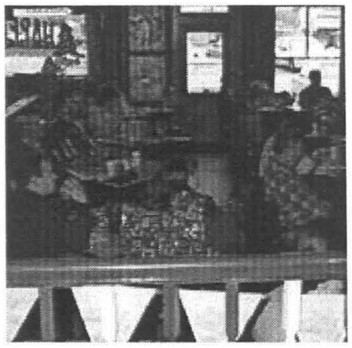


b. Stego image.

Figure 4.3. Old man in shack.

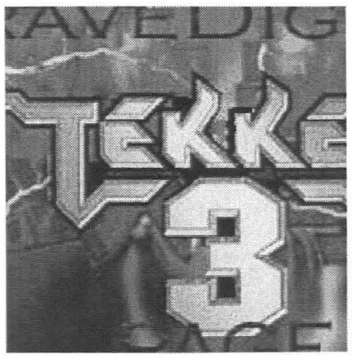


a. Original image.

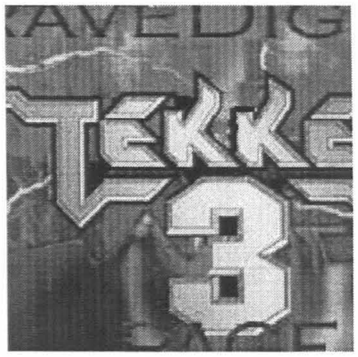


b. Stego image.

Figure 4.4. Room with people.



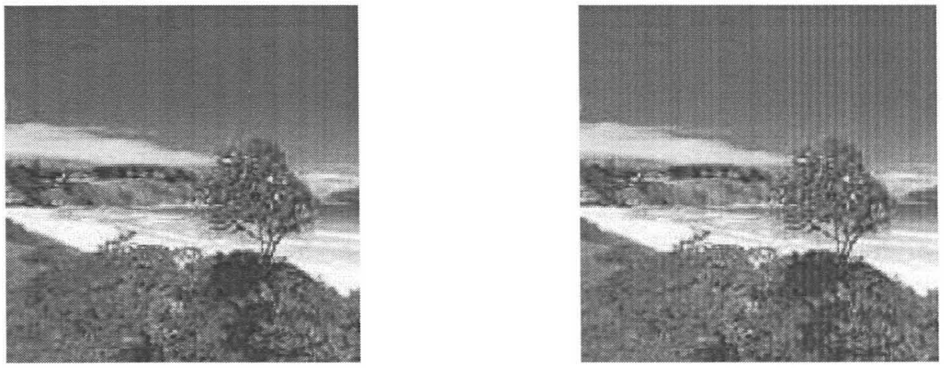
a. Original image.



b. Stego image.

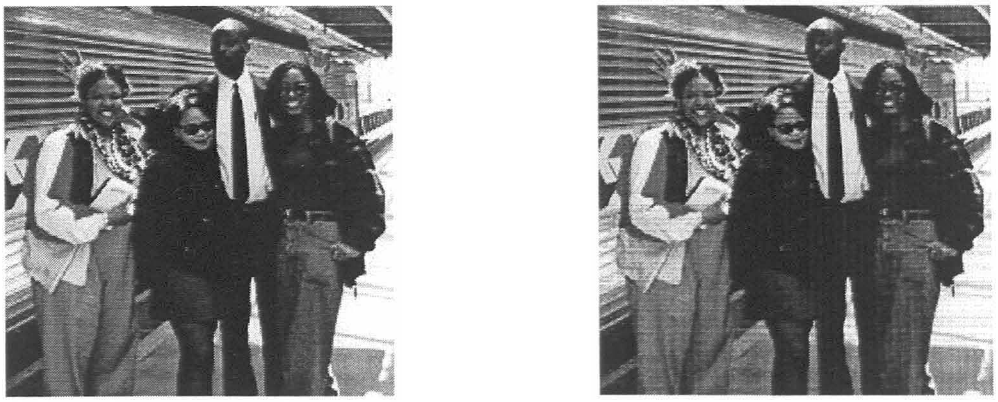
Figure 4.5. Tekken 3 logo.

المنارة للاستشارات
www.manaraa.com



a. Original image. b. Stego image.

Figure 4.6. Beach.



a. Original image. b. Stego image.

Figure 4.7. Group picture.

www.manaraa.com

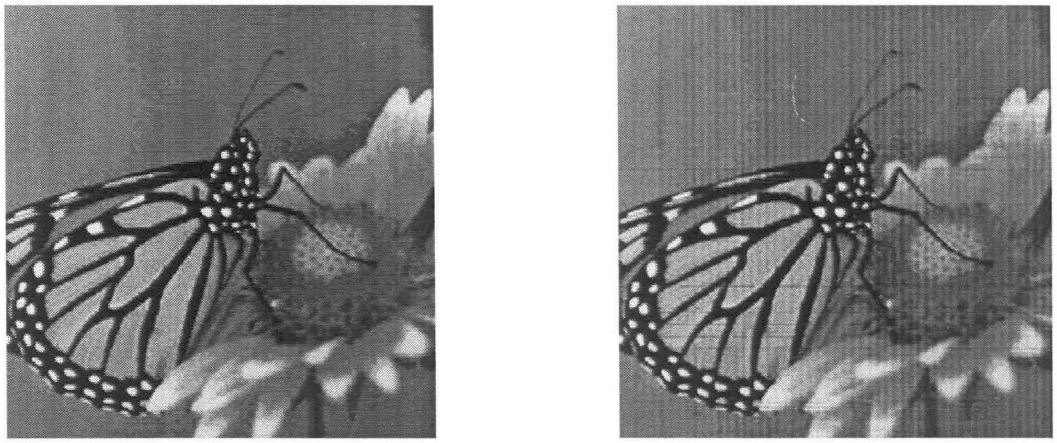
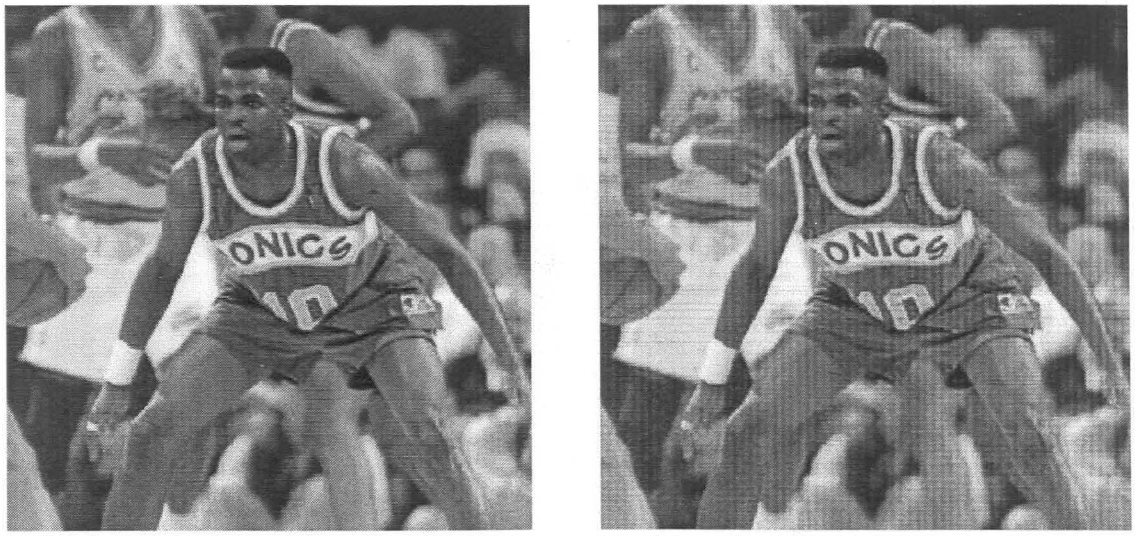

a. Original image. b. Stego image.

Figure 4.8. Butterfly.

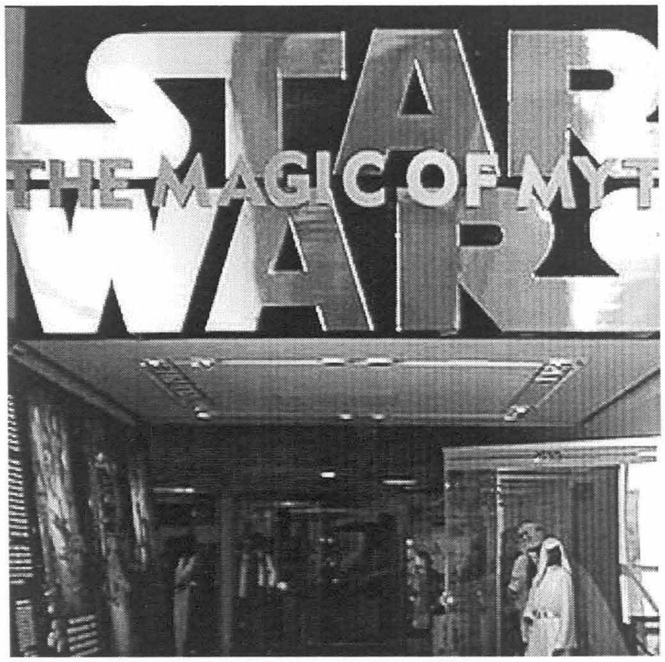


a. Original image. b. Stego image.

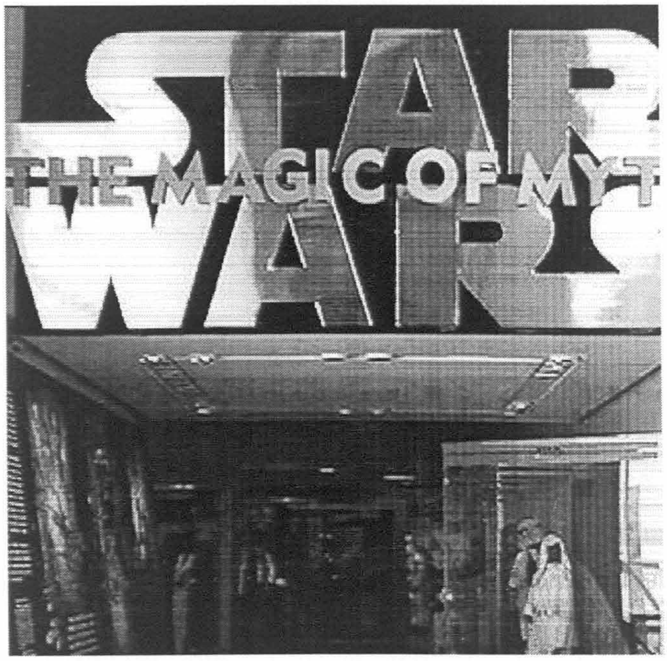
Figure 4.9. Basketball player.

للاستشارات

www.manaraa.com



a. Original image.



b. Stego image.

Figure 4.10. Star Wars©.

www.manaraa.com

CHAPTER 5: ANALYSIS AND DISCUSSION

The algorithm was run on a networked SGI workstation running IRIX. The algorithm took approximately 1.5-2 wall-clock hours for the 326 x 326 image in Figure 4.7. Looking at Figures 4.1 and 4.2, we see that the stego images appear visually close to the original images on first glance, despite their high difference measure values of 5.78 and 8.43. Indeed, the stego images almost look identical to the original images from a distance. A closer look reveals streakiness on the left side of Figure 4.1b and in the crowd of people above the left shoulder of the man in Figure 4.2b. A closer look at Figure 4.2b also reveals that much of the streakiness that exists in the image is masked by the large gray-level variations in the region of the image representing the crowd.

The stego images produced in Figures 4.3b and 4.4b are virtually indistinguishable from their original images. This is especially true in Figure 4.3b. Because the two images contain few light areas, differences between the stego and original images are difficult to see.

The stego images produced in Figures 4.5b and 4.6b are also extremely close visually to their original images. Streakiness can be seen within the “3” in Figure 4.5b, but no difference is visually perceptible in Figure 4.6b. Furthermore, the streakiness seen in Figure 4.5b would likely go unnoticed by the casual observer.

The streakiness in the stego images in Figures 4.7b, 4.8b, 4.9b and 4.10b is more apparent. Each stego image has a high difference measure (7.84, 7.31, and 9.02 respectively). In these stego images, the size of the image is important as compared to the level of reconstruction used (j=125). The larger the stego image, the more apparent

المنارة للاستشارات

www.manaraa.com

the streakiness. This is to be expected, due to the fact that a lower percentage of rows from the original images are being included in the larger stego images. For example, in Figure 4.10, 326 – 125 = 201. Thus, there are 201 rows where message data can potentially be stored. In Figure 4.1, there are only 24 such rows.

One idea for future analysis is to compare the average L_1 measure to the difference measure. There may be properties of the way an image "looks" to the eye that can be determined through comparison of these two measures.

An interesting result of our storage algorithm is the evenness in average bits per location stored in each image. Almost every image averaged between 3 and 4 bits of message data per storage location. This measure of bits per storage location could be a useful measure for future exploration of the algorithm. There does not appear to be any correlation, however, between the value of the average number of bits stored per image location and the value of the corresponding difference measure.

We believe the storage algorithm in Chapter 3 to be useful for authentication purposes. Using this algorithm, there are three keys needed to authenticate a stego image: 1) the original image, 2) the level of reconstruction j, and 3) the order in which the data was placed within the image (raster-scan, raster-scan on a rotation, etc). Without the original image and the level of reconstruction used to store data, it would be very difficult for an attacker to claim the image as their own. Even if another mark was embedded within the image via some other technique, it could be proven that the stego image was altered. Since we know which rows of the original image are included intact within any particular reconstruction and which ones are not, and since we know what was stored within a reconstruction, we have two pieces of evidence to validate our ownership of an

المنارة للاستشارات

www.manaraa.com

image. Unfortunately, this technique may not work as well as a standalone technique. Some trusted third party that could record the level of reconstruction and the original images would be necessary.

Another way this technique could be used is as an authentication tool between two parties. If a person intercepted an image being used for communication between two people and compromised it, proving that the image was compromised would not be a difficult task. Because an attacker would not know the level of reconstruction used in the storage of the message data, he or she would have to attack the stego image as a whole. However, if a recipient knows the original image and the level of reconstruction, he or she can check to make sure that rows that should be included from the original image in a particular level of reconstruction are identically reproduced in the stego image. If they are not, one can suspect an outside influence. Again, having the original image and having the level of reconstruction provide strong authentication tools for any two people trying to communicate.

One could even store multiple copies of one mark within an image at different valid locations. Suppose, for instance, an attack was initiated where the least significant bits of an image were randomly rearranged. A person can not only show that the image was altered, but may be able to recover the original data stored. Several copies of the mark would exist within the stego image that could be compared against one another. The copies of the mark could be stored at well-chosen locations, where at most one or two of the copies could be affected by such an attack, for instance. Enough copies could be embedded to prevent against the majority of those copies being affected by such an

المنارة للاستشارات

www.manaraa.com

attack. Hopefully, this method could be used to correct any errors that might have been introduced.

There is the issue of the "streakiness", or visible appearance of horizontal streaks in an image, that can be introduced using this technique. This happens because the rows of the original image are included row by row into the reconstruction, and also because the values of the stego image are, pointwise, less than the original image values, resulting in "darker" (lower) values relative to "lighter" (higher) values surrounding them. Hence, the differences between the stego image and the original image also occur row by row, resulting in horizontal streakiness in the stego image that can be seen by the eye. However, one way to solve that may be to use a different storage algorithm. The data being stored may have an effect on how streaky an image looks. If you change the storage algorithm, you may be able to eliminate much of the streakiness. For example, using a storage algorithm that alternates between rows and columns might overcome this disadvantage.

The neat thing about this algorithm is that an attacker doesn't necessarily know where the mark is stored within the image. You can pad the reconstruction with original image values in most of its locations except a few, storing your mark or multiple copies of your mark in these locations. Again, having the level of reconstruction and the original image would be important, since they tell you where you need to look for the mark or marks.

As a covert channel, this technique is not as robust, as a message can be compromised in much of the same ways as a message can be compromised if stored using any other technique. However, as a method of authentication, we believe that this

المنارة للاستشارات

www.manaraa.com

method can be very strong. It provides two levels of authentication that are difficult to get around: the fact that the original image exists and the level of reconstruction. Unfortunately, if someone had the original image, they could run a brute force attack to find which level the stegoimage was reconstructed to. However, this does not mean that they will be able to access the mark, because they don't know the storage algorithm used to hide the mark or marks.

www.manaraa.com

CONCLUSIONS AND FUTURE RESEARCH

What can this algorithm be used for? Where can this algorithm be applied? Can this algorithm be improved upon? Are there characteristics of the algorithm that have not been explored yet?

One potential use for this algorithm could be the transformation of images into covert channels. Many countries have limits on the use of cryptography, but web pages are widespread. One party could maintain a web page where he or she posts stego images, the original images coming from different locations. If the intended recipient of a message knows the locations of the original images and the level of reconstruction used for each image, he or she could read messages posted on a web page in Denmark from his or her PC in Alaska. Messages could be transmitted across the world.

Potential exists for using techniques from image processing, such as smoothening algorithms, to lessen the streaky effects seen in the stego images. Given the streakiness produced by the algorithm, the use of the algorithm alone as a watermarking technique may not be effective for some images. This could be due to the irregularities in an image, how bright or dark an image looks to the eye, or other characteristics of an image that may make our algorithm ineffective on that image. However, with digital smoothing techniques, some of the streakiness may be eliminated. There may exist a technique that could be applied to a stego image to reduce the streakiness and at the same time preserve the image quality. The only restriction on the technique would be that it would have to be a technique that was invertible. A non-invertible technique would jeopardize the ability to recover stored information from the stego image. An invertible smoothing

المنارة للاستشارات

www.manaraa.com

technique along with the algorithm may produce images that are closer to the original images.

An interesting question arises regarding the inclusion of layers in particular reconstructions. Can an image be analyzed prior to execution of the reconstruction algorithm to predict which layers will be included in a particular reconstruction and which won't? Suppose we could run an image through another algorithm to predict which layers would be included and in which order. We could then predict where streakiness would occur by looking at the rows that would not be included in a reconstruction. An image may not be useful as a covert channel if, for instance, streakiness would occur in consecutive rows, as such streakiness might be visually apparent within a stego image.

Another question arises from the discussion above. If we can predict which rows will be included within a reconstruction, can we construct images so that we know which layers will be included in which reconstruction before the layers are even computed? Can we compose images that include only the rows we choose? Instead of tailoring our message data to an image, we could then tailor an image to our message data. We could compose images that work best for the data we wish to hide. Is it even possible to manipulate an image that already exists by manipulating pixel values, for instance, to rearrange the order of layers included in a particular reconstruction? We could then take images off of the web, adjust the image before running the algorithm, and produce the best stego image for a particular reconstruction.

Still another, even more powerful application could come from the ability to compose images that included only the layers we choose. We could compose images in

www.manaraa.com

which we would not even have to hide data, but instead use the order of inclusion of rows in a reconstruction to transmit an image. Suppose our message could be translated into a decimal string, like 2 4 67 98 3 for instance. If we could compose an image that included these layers as the first 5 layers of the image, we could pass our message through the reconstruction instead of creating a stego image. We could simply post the image and tell an intended recipient the level of reconstruction to examine in order to retrieve the entire message. The message could be an encrypted message as well. We could then use the encryption algorithm as an additional key.

The major weakness of this algorithm is the streakiness produced in the stego images. The streakiness is inherent in the algorithm, as the algorithm produces reconstructions that get closer to the original image row by row as you compute higher and higher levels of reconstruction. The storage algorithm always hides one bit less of data per location than what is predicted by the *C* matrix. A better storage algorithm could reduce the streakiness produced. Another weakness of the algorithm is its image dependence. The idea of rank is discussed in [17], which provides a method of predicting what level of reconstruction will produce the original image. We saw from our results that different images contain different amounts of space for storage. This could be a strength of our algorithm, however, as more properties of the algorithm are proven. However, since certain desirable properties have not been proven to exist yet, we must assume that the image dependence is a weakness.

www.manaraa.com

REFERENCES

[1] Aura, T. Invisible Communication. EET 1995, Technical Report, Helsinki Univ. of Technology, Finland, Nov. 1995.

[2] Bender, W., Gruhl, D., Morimoto, N., and Lu, A. Techniques for Data Hiding. IBM Systems Journal,35, Nos. 3 and 4, pp. 313-335, 1996.

[3] Cox, I. J., Kilian, J., Leighton, T., and Shamoon, T. Secure Spread Spectrum Watermarking for Multimedia. Tech. Report 95-10, NEC Research Inst., Princeton, N.J., 1995.

[4] Cuninghame-Green, R. Minimax Algebra. Lecture Notes in Economics and Mathematical Systems 166. Springer-Verlag, New York, 1979.

[5] Davidson, J. L. Lattice Structures in the Image Algebra and Applications to Image Processing. Ph.D. Dissertation, University of Florida, Gainesville, FL, 1989.

[6] Fox, Barry. Hidden Watermark Traps Picture Pirates. New Scientist, 149, p. 18, Mar 2 1996.

[7] Fox, Barry. Noisy Dilemma on How To Beat Pirates. New Scientist, 150, p. 22, May 4 1996.

[8] Johnson, N. and Jajodia, S. Exploring Steganography: Seeing the Unseen. IEEE Computer, pp. 26-36, February 1998.

[9] Macq, B. and Quisquater, J. J. Cryptology for Digital TV Broadcasting. Proceedings of the IEEE, v. 83, pp. 944-57, June 1995.

[10] Moskowitz, M. Simple Timing Channels. 1994 Symposium on Security and Privacy, pp. 56-64, Oakland CA.

[11] Ritter, G. Heterogeneous Matrix Products. Image Algebra and Morphological Image Processing II, Paul D. Gader, Edward R. Dougherty, Editors, Proc. SPIE 1568, pp. 92-100, 1991.

[12] Ritter, G. and Sussner, P. The Minimax Eigenvector Transform. Image Algebra and Morphological Image Processing III, Paul D. Gader, Edward R. Dougherty, Jean C. Serra, Editors, Proc. SPIE 1769, pp. 276-282, 1992.

[13] Ritter, G. and Wilson, J.N. Handbook of Computer Vision Algorithms in Image Algebra. CRC Press, New York, 1996.

للاستشارات

www.manaraa.com

[14] Sandhu, R. Lattice-Based Access Control Models. Computer, 26, pp. 9-19, Nov. 1993.

[15] Simmons, G. J. Subliminal Channels: Past and Present. European Transactions on Telecommunications, 5, no. 4, pp. 459-473, Jul./Aug. 1994.

[16] Stallings, W. Network and Internetwork Security—Principles and Practice. Prentice Hall. Englewood Cliffs, New Jersey, 1992.

[17] Sussner, P. Matrix Decomposition in Minimax Algebra and Applications in Image Processing. Ph.D. Dissertation, University of Florida, Gainesville, FL, 1996.

[18] Wayner, P. Should Encryption Be Regulated? Byte, pp. 129-30, May 1993.

[19] zulu@iastate.edu. Chaka Allen's homepage router. Chaka Allen, http://www.public.iastate.edu/~zulu, April 23 1998.

www.manaraa.com